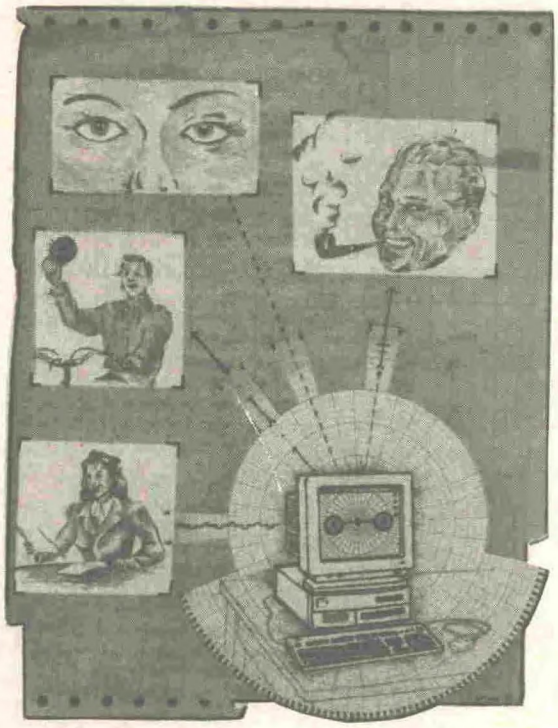
เทคนิคการเขียนโปรแกรมด้วย VB4



สนุกกับเทคนิคต่าง ๆ ในการเขียนโปรแกรมด้วย VB4

วิชวลเบสิกเป็นภาษาที่ช่วยให้หลายคนที่ฝันจะเขียนโปรแกรมบนวินโดวส์นั้นเป็นจริงขึ้นมาได้ โดยเฉพาะเมื่อมีวิชวลเบสิก 4.0 ที่ช่วยให้เราสร้างแอพพลิเคชันบนวินโดวส์ 95 ได้อย่างง่ายดาย ไม่เฉพาะแต่วิชวลเบสิกเท่านั้นที่มีเทคนิคในการเขียนโปรแกรมให้มีประสิทธิภาพสูง เพราะทุก ๆ ภาษาก็มักมีเทคนิคพิเศษเหล่านี้เช่นกัน แต่ก็โชคดีที่วิชวลเบสิกนั้นได้รับความนิยมสูงอยู่ในปัจจุบัน จึงทำให้มีข้อมูลที่เกี่ยวกับวิชวลเบสิกเผยแพร่อยู่บนอินเทอร์เน็ตมากพอสมควร เทคนิคต่าง ๆ ที่ผู้เขียนจะกล่าวถึงในบทความนี้จะเน้นเฉพาะวิชวลเบสิก 4.0 เท่านั้น โดยผู้เขียนได้ดัดแปลงมาจากข้อมูลในเวบไซท์ www.apexsc.com/vb/davem/vbtt.html อย่างไรก็ตาม ผู้อ่านก็สามารถดัดแปลงไปใช้ให้เกิดประโยชน์ได้ถ้าใช้วิชวลเบสิกรุ่นอื่น การนำเสนอเรื่องราวในบทความนี้จะแยกกล่าวถึงเป็นหัวข้อ ๆ ไป ขอให้สนุกกับการเขียนโปรแกรมนะครับ

การเรียกใช้ซีเคียวดาต้าเบสจากวิชวลเบสิก

ปกติการจะเรียกใช้ซีเคียวดาต้าเบสที่สร้างจากแอ็กเซสนั้น ไม่มีวิธีการที่กล่าวถึงอย่างแน่นอนสำหรับการทำงานแบบ 16 บิต หรือ 32 บิต ดังนั้นผู้เขียนโปรแกรมจึงมักรู้สึกสับสนพอสมควรเมื่อจำเป็นต้องใช้งาน โดยเฉพาะเมื่อต้องใช้งานกับเอนจิน Jet 2.5 แบบ 16 บิต ซึ่งมีบักที่จะเกิดขึ้นเมื่อเรียกเมธอด CreateWorkspace จนทำให้ระบบมีปัญหา หากผู้ใช้ไม่กำหนดค่าในพรอพเพอร์ตี้ส์ DefaultUser และ DefaultPassword ที่ตรงกับที่ซีเคียวดาต้าเบสต้องการไว้ก่อน อย่างไรก็ตาม ปัญหานี้จะไม่เกิดขึ้นหากปฏิบัติตามแนวทางการใช้ซีเคียวดาต้าเบสดังนี้

สำหรับโปรแกรมที่ทำงานแบบ 16 บิต ผู้อ่านต้องสร้างไฟล์ชื่อ APPNAME.INI ขึ้นมาก่อน โดยชื่อ APPNAME นั้นจะต้องเปลี่ยนแปลงไปตามชื่อไฟล์โปรแกรม .EXE ที่ผู้อ่านกำหนดไว้ ภายในไฟล์จะต้องใส่ข้อมูลเหล่านี้ไว้

```
[Data]
Database=D:\PATH\DBNAME.MDB
[Options]
SystemDB=D:\PATH\SYSTEM.MDA
```

จากนั้นผู้อ่านก็พร้อมที่จะเปิดดาต้าเบสดังนี้

```
Dim sUserName As String
Dim sPassword As String
Dim db As Database
Dim ws As Workspace
      sUserName = "Your Name Here"
      sPassword = "Bite_Me"

      With DBEngine
            .IniPath =
"D:\PATH\APPNAME.INI"
            .DefaultUser = sUserName
            .DefaultPassword =
sPassword
      End With

      Set ws =
DBEngine.CreateWorkspace("Name",
sUserName, sPassword)

      Set db =
ws.OpenDatabase("D:\PATH\DBNAME.MDB"...)
```

ส่วนโปรแกรมที่ทำงานแบบ 32 บิตนั้นไม่จำเป็นต้องสร้างไฟล์ .INI และไม่ต้องกำหนดค่า DefaultUser และ DefaultPassword ด้วย ผู้อ่านเพียงแต่กำหนดพรอพเพอร์ตี้ส์ SystemDB ของ DBEngine ให้ชี้ไปยังดาต้าเบสที่ต้องการดังนี้

```
Dim sUserName As String
Dim sPassword As String
Dim db As Database
Dim ws As Workspace
      sUserName = "Your Name Here"
      sPassword = "Bite_Me"

      DBEngine.SystemDB = "D:\PATH\SYSTEM.MDW"
      Set ws = DBEngine.CreateWorkspace("Name",
sUserName, sPassword)

      Set db =
ws.OpenDatabase("D:\PATH\DBNAME.MDB"...)
```

การจัดขนาดและตำแหน่งของคอนโทรล

กรณีที่มีคอนโทรลชนิดหนึ่ง เช่นคอมมานด์บัททอน ซึ่งมีขนาด (ความกว้างและสูง) ตำแหน่ง (แนวตั้งและนอน) ไม่เท่ากันวางอยู่บนฟอร์ม หากผู้อ่านต้องการกำหนดให้พรอพเพอร์ตี้ส์บางอย่างของบัททอนมีค่าเท่ากัน เช่นต้องการจัดให้ขอบด้านซ้ายของบัททอนวางอยู่ในแนวเดียวกัน ก็ให้จับบัททอนเหล่านั้นรวมเป็นกลุ่มเดียว โดยวาดกรอบล้อมรอบบัททอนบนฟอร์มที่ต้องการให้อยู่ภายในกรอบเดียวกัน จากนั้นเลือกหน้าต่าง Properties และคลิ๊กเม้าส์ที่พรอพเพอร์ตี้ส์ Left ทั้งหมด 3 ครั้ง บัททอนเหล่านั้นจะถูกจัดให้อยู่ในแนวเดียวกันทันที หรือหากต้องการปรับให้ขนาดความกว้างหรือสูงของบัททอนเท่ากันทั้งกลุ่มก็ให้คลิ๊กที่พรอพเพอร์ตี้ส์ Width หรือ Height แทน

การจัดการ Error แบบง่าย ๆ

วิธีที่จะเขียนโค้ดเพื่อจัดการกับ Error ที่จะเกิดขึ้นเมื่อเขียนโปรแกรมด้วยวิชวลเบสิกแบบง่าย ๆ สามารถทำได้ดังนี้

```
Option Explicit

Global nReturn As Integer
Global cMsg As String
Global Const errExit = 0
Global Const errResume = 1
Global Const errNext = 2
Global Const errSelect = 3

Public Function errHandler(nErrorNumber As
Integer, cErrText As_ String, nErrOption As
Integer) As Integer

cMsg = ""
cMsg = cMsg + "Err:" + Chr(9) + Str(nErrorNumber)
+ Chr(13)
cMsg = cMsg + "Text:" + Chr(9) + cErrText +
Chr(13)

 ErrWriteLogFile cMsg

Select Case nErrOption
    Case Is = errExit
        MsgBox cMsg, vbCritical, "Exiting
program..."
        GoTo errHandlerEnd
    Case Is = errResume
        MsgBox cMsg, vbCritical, "Error"
        errHandler = errResume
    Case Is = errNext
        MsgBox cMsg, vbCritical, "Error"
        errHandler = errNext
    Case Is = errSelect
        nReturn = MsgBox(cMsg, vbCritical +
vbAbortRetryIgnore,_ "Error")
        Select Case nReturn
        Case Is = vbAbort
           GoTo errHandlerEnd
        Case Is = vbRetry
            errHandler = errResume
        Case Is = vbIgnore
            errHandler = errNext
        End Select
 End Select
 Exit Function

errHandlerEnd:
    MsgBox "Click OK to Exit Program"
    End
End Function

Public Sub ErrWriteLogFile(cLogMsg)

On Error GoTo errWriteLogFileErr

    Dim cFile$
    Dim nFile
    Dim ErrDir$
    ErrDir = App.Path

    nFile = FreeFile()
    cFile = ErrDir + Format(Now, "mmddyyhhss") +
".err"
    Open cFile For Output As nFile
    Print #nFile, "***ERROR REPORT - [" +
App.EXEName + "] ***"
    Print #nFile, ""
    Print #nFile, "DATE: " + Format(Now, "General
Date")
    Print #nFile, ""
    Print #nFile, cLogMsg
    Print #nFile, ""
    Print #nFile, "***eof***"

    Close #nFile
    GoTo errWriteLogFileExit

errWriteLogFileErr:
    MsgBox Str(Err) + "-" + Error$, vbCritical,
"Unable to Write Error Log"
        Exit Sub
errWriteLogFileExit:

End Sub
```

■

โดยที่ตอนเรียกใช้ให้เขียนโค้ดในซับรูทีนหรือฟังก์ชันดังนี้

```
Private Sub errHTest_Click()
On Error GoTo ErrHtestError:

' Your Code Here

GoTo errHTestExit
ErrHtestError:
    nReturn = errHandler(Err, Error$, errNext)
    Resume Next

errHTestExit:
    Exit Sub

End Sub
```

วิธีกำหนดให้เลือกข้อความโดยอัตโนมัติ

เมื่อใดที่ผู้ใช้คลิ๊กเม้าส์เพื่อเลือกเท็กซ์บ็อกซ์หรือคอมโบบ็อกซ์ที่อยู่บนฟอร์ม เราควรที่จะเซตให้ข้อความที่อยู่ภายในเท็กซ์บ็อกซ์นั้นอยู่ภายในแถบสีทันที ผู้อ่านอาจเขียนซับรูทีนง่าย ๆ ไว้ใช้งานดังนี้

```
Sub Highlight()

    On Error Resume Next

    Dim FRM1 As Form
    Set FRM1 = Screen.ActiveForm

    FRM1.ActiveControl.SelStart = 0
    FRM1.ActiveControl.SelLength =
Len(FRM1.ActiveControl)

End Sub
```

จากนั้นให้เรียกใช้ซับรูทีน Highlight ในเหตุการณ์ GotFocus เพื่อเซตให้ข้อความเป็นแถบสีทันทีที่โฟกัสคอนโทรล

จัดฟอร์มให้อยู่ตรงกลาง

ปกติก่อนที่จะมีการปรากฏตัวของฟอร์มเรามักต้องจัดตำแหน่งของฟอร์มให้อยู่ตรงกลางบนจอภาพ หรือกลางฟอร์มที่ใหญ่กว่าเสมอ ดังนั้นเราจะสามารถเขียนซับรูทีน CenterForm เพื่อทำหน้าที่จัดตำแหน่งฟอร์มเทียบกับออบเจ็กต์ที่อ้างอิงได้ดังนี้

1. CenterForm Main, Screen ใช้เรียกเมื่อต้องการจัดให้ฟอร์มหลักอยู่กลางจอภาพ

2. CenterForm Test, Screen, vbModal ใช้เรียกเมื่อต้องการจัดฟอร์มย่อยแบบ modal ให้อยู่กลางจอภาพ

3. CenterForm Main, Info ใช้เรียกเมื่อ Info เป็นชื่อของคอนโทรล SysInfo โดยผู้อ่านต้องการจัดฟอร์มหลัก Main ให้อยู่ตรงกลางสกรีนวิวเอเรียล, ปรับขนาดและตำแหน่งของวินโดวส์ทาสก์บาร์

4. CenterForm Test, Main ใช้เรียกเมื่อต้องการจัดให้ฟอร์ม Test อยู่ตรงกลางฟอร์มหลัก

```
Sub CenterForm(Child As Object, Parent As
Object, Optional LOffset As Variant,_ Optional
TOffset As Variant, Optional vMode As Variant)
    Dim Left%, Top%, Mode%, L_Offset%, T_Offset%

    If TypeOf Parent Is SysInfo Then
        Left% = Parent.WorkAreaLeft +
(Parent.WorkAreaWidth - Child.Width) / 2
        Top% = Parent.WorkAreaTop +
(Parent.WorkAreaHeight - Child.Height) / 2
    ElseIf TypeOf Parent Is MDIForm Then
        If Child.MDIChild = True Then
            Left% = (Parent.ScaleWidth -
Child.Width) / 2
            Top% = (Parent.ScaleHeight -
Child.Height) / 2
        Else
            Left% = Parent.Left + (Parent.Width
- Child.Width) / 2
            Top% = Parent.Top + (Parent.Height -
Child.Height) / 2
        End If
    ElseIf TypeOf Parent Is Screen Then
        Left% = (Parent.Width - Child.Width) / 2
        Top% = (Parent.Height - Child.Height) /
2
    ElseIf TypeOf Parent Is Form Then
        If Parent.MDIChild = True Then
            Left% = Parent.Left + (Parent.Width
- Child.Width) / 2
            Top% = Parent.Top + (Parent.Height -
Child.Height) / 2
            Dim I%
            For I% = 0 To Forms.Count - 1

                If TypeOf Forms(I%) Is MDIForm
Then
                    Left% = Left% +
(Forms(I%).Width - Forms(I%).ScaleWidth)
/ 2 + Forms(I%).Left
                    Top% = Top% +
(Forms(I%).Height - Forms(I%).ScaleHeight)
/ 2 + Forms(I%).Top
                    Exit For
                End If
            Next I%
        Else
            Left% = Parent.Left + (Parent.Width
- Child.Width) / 2
            Top% = Parent.Top + (Parent.Height -
Child.Height) / 2

        End If
    Else
        Exit Sub
    End If
    If IsMissing(vMode) Or Child.MDIChild = True
Then Mode% = vbModeless
Else Mode% = Int(vMode)
    If IsMissing(LOffset) Then L_Offset% = 0
Else L_Offset% = Int(LOffset)
    If IsMissing(TOffset) Then T_Offset% = 0
Else T_Offset% = Int(TOffset)
    Child.Move Left% + L_Offset%, Top% +
T_Offset%
    Child.Show Mode%
End Sub
```

เปลี่ยนเม้าส์พอยน์เตอร์ด้วยคลาส

เราสามารถเปลี่ยนรูปพอยน์เตอร์ของเม้าส์ให้เป็นรูปนาฬิกาทรายและเปลี่ยนกลับไปเป็นรูปก่อนหน้าเองเมื่อโพรซีเดอร์นั้นทำงานจบหรือ

หยุดไปกลางคัน โดยการใช้งานต้องเขียนโค้ดในซับรูทีนที่จะใช้ดังนี้

```
Sub AProcedure()
   Dim Hourglass As New HourglassObject
   Hourglass.DisplayCursor
  ' *** your procedure code here...
End Sub
```

สำหรับโค้ดของคลาสที่จะต้องเขียนมีดังนี้

```
BEGIN
  MultiUse = -1  'True
END
Attribute VB_Name = "clsHourglass"
Attribute VB_Creatable = True
Attribute VB_Exposed = True
Option Explicit

Private mMousePointer   As Long

Public Sub DisplayCursor(Optional
SubstitutePointer As Variant)
   mMousePointer = Screen.MousePointer
   If IsMissing(SubstitutePointer) Then
      Screen.MousePointer = vbHourglass
   Else
      Screen.MousePointer = SubstitutePointer
   End If
End Sub

Private Sub Class_Terminate()
   ' *** Restore pointer ***

  Screen.MousePointer = mMousePointer
End Sub
```

แปลงชื่อไฟล์แบบยาวให้เป็นแบบสั้น

การใช้ชื่อไฟล์แบบสั้นบางครั้งยังมีความจำเป็นอยู่เมื่อต้องส่งชื่อไฟล์เป็นพารามิเตอร์ให้กับโปรแกรมแบบเก่าที่ทำงานบนดอส ในตัวอย่างนี้ให้สร้างคอนโทรล DirListBox ชื่อ dirLongName ไว้บนฟอร์ม จากนั้นสร้างเท็กซ์บ็อกซ์ชื่อ txtShortName ไว้บนฟอร์มด้วย จากนั้นเขียนโค้ดต่อไปนี้ไว้ เมื่อใดที่คลิ๊กเม้าส์บน DirListBox ก็จะมีชื่อไฟล์สั้นแสดงให้เห็นด้วยพร้อมกัน

```
Option Explicit

Private Declare Function GetShortPathName Lib
"KERNEL32" Alias "GetShortPathNameA" (ByVal
lpszLongPath As String, ByVal lpszShortPath As
String, ByVal cchBuffer As Long) As Long

Public Function GetShortFileName(ByVal FileName
As String) As String
    Dim rc As Long
    Dim ShortPath As String
    Const PATH_LEN& = 164

    ShortPath = String$(PATH_LEN + 1, 0)
    rc = GetShortPathName(FileName, ShortPath,
PATH_LEN)
    GetShortFileName = Left$(ShortPath, rc)
End Function

Private Sub dirLongName_Click()
    Dim LongName As String
    LongName =
dirLongName.List(dirLongName.ListIndex)
    txtShortName = GetShortFileName(LongName)
End Sub

Private Sub Form_Load()
    dirLongName_Click
End Sub
```

การสร้างไฟล์ชั่วคราว

มีการเรียกใช้ฟังก์ชันเอพีไอ 32 บิต ที่จะสร้างไฟล์ชั่วคราวขึ้นเองในไดเรคทอรี TEMP ของวินโดวส์ การเรียกใช้ฟังก์ชันนี้ก็เพียงแต่ส่งชื่อที่จะใช้นำหน้าชื่อไฟล์เท่านั้น เช่นถ้าเรียกฟังก์ชัน sGetTempFile ("VB") ก็จะมีการสร้างไฟล์ชื่อ E:\WIN95\TEMP\VB9295.TMP ให้

```
Declare Function OSGetTempPath& Lib "kernel32"
Alias "GetTempPathA" (ByVal BufferLength&, ByVal
Result$)
Declare Function OSGetTempFilename& Lib
"kernel32" Alias "GetTempFileNameA" (ByVal
FilePath$, ByVal Prefix$, ByVal wUnique&, ByVal
TempFileName$)

Function sGetTempFile(ByRef sPrefix As String)
Dim sFilePath As String
Dim sTempResult As String
Dim lCharCount As Long

   Const MAX_RETURN = 3000

   sTempResult = Space$(MAX_RETURN)
   lCharCount = OSGetTempPath&(MAX_RETURN,
sTempResult)

   sFilePath = Left$(sTempResult, lCharCount)

   sTempResult = Space$(MAX_RETURN)
   lCharCount = OSGetTempFilename&(sFilePath,
sPrefix, 0, sTempResult)
   sGetTempFile = Left$(sTempResult, lCharCount)

End Function
```

การสร้างวินโดวส์ 95 เชลล์ลิงค์

ถ้าผู้อ่านเคยดูซอร์สโค้ดของ Setup Kit (SETUP1.VBP) ของวิชวลเบสิก 4.0 ผู้อ่านอาจเห็นการเรียกใช้ STKIT432.DLL เพิ่มสร้างชอตคัตให้กับเมนู Start ของวินโดวส์ 95 ดังนี้

```
Declare Function fCreateShellLink Lib
"STKIT432.DLL" (ByVal lpstrFolderName As
String, ByVal lpstrLinkName As String, ByVal
lpstrLinkPath As String, ByVal lpstrLinkArgs As
String) As Long
```

เราสามารถนำวิธีการดังกล่าวไปใช้สร้างชอตคัตแบบนี้ได้เองเช่นกัน โดย lpstrFolderName ใช้อ้างถึงโฟลเดอร์ของเมนู Start โดยทั่วไปคือ Programs ซึ่งอยู่ที่ \Windows\Start Menu\Programs ดังนั้น

หากผู้อ่านส่งค่า null ไปชอตคัตจะถูกสร้างในโฟลเดอร์ของตัวโปรแกรมเอง หรือจะกำหนดไปยังไดเรคทอรีใดก็ได้บนฮาร์ดดิสก์ เช่น หากต้องการส่งไปยังเดสก์ท็อปของผู้ใช้ก็เขียนโค้ดดังนี้

```
lResult = fCreateShellLink("..\..\Desktop",
"Shortcut Title", "d:\path\appname_ .exe", "")
```

แดร็กและดร็อบจาก Win95 Explorer

เราสามารถเขียนโปรแกรมให้รับทราบชื่อไฟล์ที่แดร็กจาก Win95 Explorer แล้วนำมาดร็อบไว้ในแอพพลิเคชันที่เขียนได้ง่าย ๆ ดังนี้

```
Type POINTAPI
        x As Long
        y As Long
End Type

Type MSG
    hWnd As Long
    message As Long
    wParam As Long
    lParam As Long
    time As Long
    pt As POINTAPI
End Type

Declare Sub DragAcceptFiles Lib "shell32.dll"
(ByVal hWnd As Long, ByVal fAccept As Long)
Declare Sub DragFinish Lib "shell32.dll" (ByVal
hDrop As Long)
Declare Function DragQueryFile Lib "shell32.dll"
Alias "DragQueryFileA" (ByVal hDrop As Long,
ByVal UINT As Long, ByVal lpStr As String, ByVal
ch As Long) As Long
Declare Function PeekMessage Lib "user32" Alias
"PeekMessageA" (lpMsg As MSG, ByVal hWnd As
Long, ByVal wMsgFilterMin As Long, ByVal
wMsgFilterMax As Long, ByVal wRemoveMsg As Long)
As Long

Public Const PM_NOREMOVE = &H0
Public Const PM_NOYIELD = &H2
Public Const PM_REMOVE = &H1
Public Const WM_DROPFILES = &H233

Public Sub WaitForDrop()

Dim FileDropMessage As MSG       ' Msg Type
Dim FileDropped     As Boolean   ' True if Files
where dropped
Dim hDrop           As Long      ' Pointer to the
dropped file structure
Dim Filename        As String * 128  ' the
dropped filename
Dim NumOfDroppedFiles As Long    ' the amount of
dropped files
Dim CurFile         As Long      ' the current
file number

Do
    FileDropped = PeekMessage(FileDropMessage,
0, WM_DROPFILES, WM_DROPFILES, PM_REMOVE Or
PM_NOYIELD)
    If FileDropped Then
        hDrop = FileDropMessage.wParam
        NumOfDroppedFiles = DragQueryFile(hDrop,
True, Filename, 127)
        Form1.List1.Clear
        For CurFile = 1 To NumOfDroppedFiles
            ret% = DragQueryFile(hDrop, CurFile -
1, Filename, 127)
            Form1.lblNumDropped =
LTrim$(Str$(NumOfDroppedFiles))
            Form1.List1.AddItem Filename
        Next CurFile
        DragFinish (hDrop)
    End If
    DoEvents
Loop
End Sub
```

ก่อนที่จะใช้งานได้ต้องเรียกฟังก์ชันเอพีไอ DragAcceptFiles เพื่อแจ้งชื่อฟอร์มให้ทราบก่อนใช้ดังนี้

```
DragAcceptFiles Form1.hWnd, True
```

ข้อบกพร่องของเอกสาร CompactDatabase

ในคู่มือของวิชวลเบสิก 4.0 ให้ข้อมูลเกี่ยวกับเมธอด CompactDatabase ว่าต้องการพารามิเตอร์ 4 ตัวดังนี้

```
DBEngine.CompactDatabase OldDatabase,
NewDatabase [, Locale [,Options]]
```

แต่มีผู้รายงานปัญหาที่เกิดจากการใช้งานเมธอดนี้กับดาต้าเบสแบบที่มีพาสเวิร์ด ทางแก้ไขปัญหาเมื่อใช้ Object Browser ตรวจสอบปรากฏว่าเมธอด CompactDatabase ต้องการพารามิเตอร์ 5 ตัวดังนี้

```
DBEngine.CompactDatabase SrcName, DstName,
DstConnect, Options, SrcConnect
```

โดยต้องเซตให้พารามิเตอร์ตัวสุดท้ายเป็น ";pwd=password" จึงสามารถทำงานต่อได้

แก้ปัญหาข้อบกพร่อง 3020 และ 3426 ของดาต้าคอนโทรล

ถ้าผู้อ่านเขียนโค้ดต่อไปนี้ใช้ในวิชวลเบสิก 3.0 ก็คงจะไม่มีปัญหาอะไร

```
Private Sub cmdUpdate_Click()
    'Save contents of bound controls
    'to underlying recordset
    datCtl.Recordset.Update
End Sub
```

แต่ถ้าเกิดไปเขียนใช้ในวิชวลเบสิก 4.0 รุ่น 16 บิตก็จะเกิดรันไทม์เออเรอร์ Error 3426 — "The action was cancelled by an associated

object" หรือในรุ่น 32 บิต ก็อาจเกิด Error 3020 — "Update or CancelUpdate without AddNew or Edit" ได้ ปัญหานี้ดูเหมือนจะเกิดจากในวิชวลเบสิก 4.0 ไม่ได้จัดการทำงานกับเมธอด Edit อย่างชัดเจน เมื่อดาต้าคอนโทรลเลื่อนไปยังเรคอร์ดใหม่ ทางแก้ไขจะต้องตรวจสอบที่ EditMode ของเรคอร์ดเซต และจัดการให้เรียบร้อยดังนี้

```
Private Sub cmdUpdate_Click()
      If datCtl.Recordset.EditMode = dbEditNone
Then
         datCtl.Recordset.Edit
      End If
      datCtl.Recordset.Update
End Sub
```

หรือจะทดแทนเมธอด Update ด้วยเมธอด UpdateRecord ของดาต้าคอนโทรล ซึ่งเทียบเท่ากับการเรียกใช้ Edit ตามหลัง Update

ทำอย่างไรหากต้องการยกเลิก AddNew/Edit

ในคู่มือของวิชวลเบสิกกล่าวถึงเมธอด AddNew และ Edit ว่าในรุ่นโปรเฟสชันนัล ถ้าผู้ใช้ไม่ได้ใช้ดาต้าคอนโทรลและใช้เมธอด Find หรือ Move หรือปิดเรคอร์ดเซตขณะที่กำลังทำ AddNew หรือ Edit การแก้ไขข้อมูลที่ทำไปจะไม่เกิดขึ้น ดังนั้นวิธียกเลิกการแก้ไขข้อมูลและเลื่อนตำแหน่งจากเรคอร์ดปัจจุบัน แต่ก็มีวิธียกเลิกการแก้ไขโดยไม่เลื่อนไปจากเรคอร์ดปัจจุบันซึ่งสามารถทำได้ดังนี้

```
' In VB4...
Dim rs As Recordset

' Begin an edit
rs.Edit
...
' Cancel it
rs.Move 0

' In VB3...
Dim ds As Dynaset
Dim sBkMk As String

' Save bookmark of current record
sBkMk = ds.Bookmark
' Begin an edit
ds.Edit
...
' Cancel it
ds.Bookmark = sBkMk
```

เวบไซท์ยอดนิยม

(ต่อจากหน้า 226)

41. http://www.sonicnet.com
42. http://www.word.com
43. http://www.winmag.com
44. http://www.slip.net
45. http://www.earthlink.net
46. http://www.conner.com
47. http://www.mit.edu
48. http://www.pbs.org
49. http://www.cool.infi.net
50. http://www.postcards
51. http://www.media.mit.edu
52. http://www.euro.net
53. http://www.york.ac.uk
54. http://www.stpt.com
55. http://www.nasa.gov
56. http://www.metro.turnpike.net
57. http://www.worldvillage.com
58. http://www.mind.net
59. http://www.bev.net
60. http://www.us.dell.com
61. http://www.ultimatetv.com
62. http://www.sln.fi.edu
63. http://www.cyberzine.com
64. http://www.netgen.com
65. http://www.dtd.com
66. http://www.turnpike.net
67. http://www.submit-it.com
68. http://www.directory.net
69. http://www.realitycom.com
70. http://www.virtualynx.com
71. http://www.eclipse.net
72. http://www.ibic.com
73. http://www.thoughtport.com
74. http://www.prgone.com
75. http://www.magellan.mckinley.com
76. http://www.globall.com
77. http://www.inktomi.berkeley.edu
78. http://www.yellow.com
79. http://www.newtoo.manifest.com
80. http://www.eshnav.com
81. http://www.chico.rice.edu
82. http://www.xplore.com
83. http://www.netsurf.com
84. http://www.uroulette.com
85. http://www.tnet.de
86. http://www.iplex.com

และ ฯลฯ